

1. Scope The textually first use of a name, whether of a procedure, function label or variable (whether DIMed or not) defines the names type. Any subsequent attempt to redefine this name is an error.
2. STOP with a message. If a message is printed then no line number is printed.
3. RANDOMISE This function allows RND to begin with an arbitrary seed value.
4. Error in Production 136 Remove all references to functions.
5. NEXT Variable Make the control variable in NEXT optional. System provides it by default.
6. Integer Type Add the integer type to the kernel.
7. Formatted PRINT

Existing definition

Normally the variables are printed out using a system-defined format for the particular type of variable. If the USING <format info> part is present this is not the case. Instead the format is determined from the string in the format clause. The meaning of the characters in the string of the <format info> is as follows: A substring of one or more embedded hash signs (#), optionally containing a single point (.), will be substituted on output by the value of a corresponding numeric variable. The variable will be printed out with as many digits precision as specified by hash signs to the right of the point. If no point is specified, no fractional part is printed. Leading zeroes in the integer part are replaced by blanks; trailing zeroes in the fractional part are printed. If the variable has an integer part which is too large to be represented in the field, the field is printed out as all hash signs.

Propose to add :

A substring of one or more contiguous dollar signs (\$) will be substituted on output by the value of a corresponding string variable. If the string variable is shorter than the format substring, it will be left justified in the field, and blank filled on the right. If the string variable is longer than the field, then as much as possible of the variable will be printed out within the field,

starting on the left. If it is required to print out one of the special characters hash or dollar without any substitution, it must be preceded by a (^). If a hat itself is printed, then two successive hats should be written. Any other character is simply printed out as it stands.

Special format substrings and variables are matched up from left to right. The first format substring defines the format for the first variable in the variable list, and so on. If a format substring is inappropriate to the type of the corresponding variable, an error will occur.

\$